

TRABAJO PRÁCTICO N°2

Ingeniería de Software

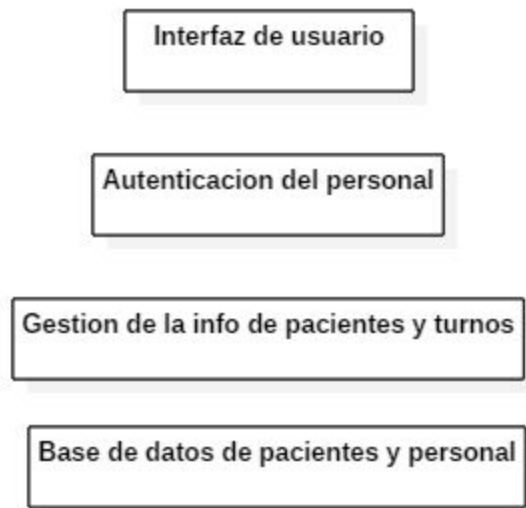
Arquitectura y Diseño del Sistema

Funes Joaquin
Pardina Lucas
Carpio Ramiro

Arquitectura del Sistema

Patrón de Arquitectura

Para nuestro sistema se optó por el patrón de arquitectura en capas, que es una buena forma de lograr separación e independencia. Aquí, la funcionalidad del sistema está organizada en capas separadas, y cada una se apoya en las facilidades y servicios ofrecidos por la capa que está inmediatamente debajo de ella. A continuación se muestra la arquitectura de nuestro sistema.



En la capa inferior se incluye el software de soporte al sistema, en nuestro caso es el soporte de la base de datos con todos los datos asociados a los pacientes y el personal hospitalario. La siguiente capa implementa la funcionalidad del sistema, que para nosotros es todo lo relacionado a la gestión y control de los pacientes como así también de los turnos del hospital. La tercera capa se relaciona con la autenticación y verificación tanto del personal administrativo como de los doctores, para que cada cual realice las operaciones permitidas de acuerdo a su rol. La capa superior es la encargada de la interfaz de usuario.

Relaciones entre capas:

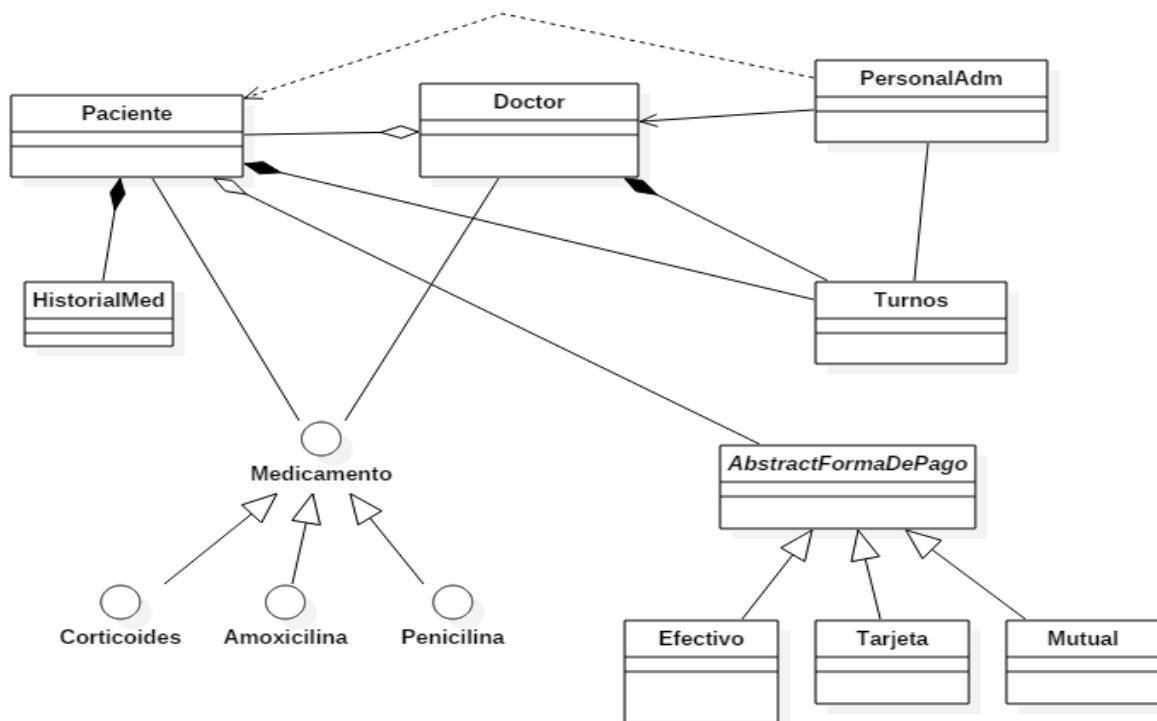
- **Base de datos de pacientes y personal - Gestión de la info de pacientes y turnos:** Estas dos capas se relacionan de forma que la capa inferior le proporciona a la siguiente todos los datos y métodos sobre personal para que ésta pueda crear, modificar y borrar los Turnos y/o Pacientes.
- **Gestión de la info de pacientes y turnos - Autenticación del personal:** Aquí, la capa inferior brinda los métodos necesarios para que en la capa que le sigue se puedan autenticar tanto los Doctores como el PersonalAdm.
- **Autenticación del personal - Interfaz de usuario:** Una vez que un personal se autentifica, sea Doctor o PersonalAdm, se procederá a la interfaz gráfica correspondiente.

Casos de Prueba de Integración

- **Caso I:** Verificar integración entre Doctor y Paciente, comprobando la correcta asignación de medicamentos, la creación del HistorialMed y el aviso sobre alergias a medicamentos.
- **Caso II:** Verificar integración entre Doctor y PersonalAdm, comprobando la correcta actualización de la lista de turnos de un Doctor con la lista de turnos del PersonalAdm.
- **Caso III:** Verificar integración entre Turnos y PersonalAdm, comprobando la correcta creación de turnos con sus respectivo paciente y doctor.

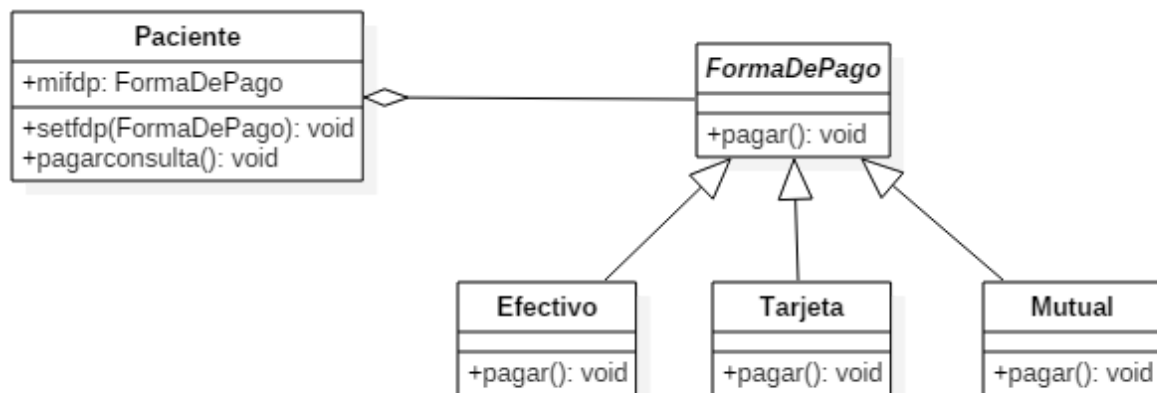
Diseño del Sistema

Diagrama de Clases



Patrones de Diseño

Se hace uso del patrón de diseño **Strategy** para controlar la forma de pago de las consultas médicas, ya que dichas consultas podrán ser abonadas mediante efectivo, tarjeta de crédito o por mutual médica.



El patrón **Observer** está implementado mediante las clases abstractas **PersonalAdm** y **Doctor**, y de ésta manera cuando el personal administrativo cree un nuevo turno le notificará al doctor que hay un nuevo turno en su agenda. En nuestro caso la clase **PersonalAdm** es el sujeto abstracto y el encargado de proporcionar la interfaz para añadir y quitar observadores, **Doctor** sería el observador abstracto. Los observadores concretos que serían **DoctorA** y **DoctorB** mantienen una referencia al sujeto concreto (**Personal1**) y una copia de su estado.

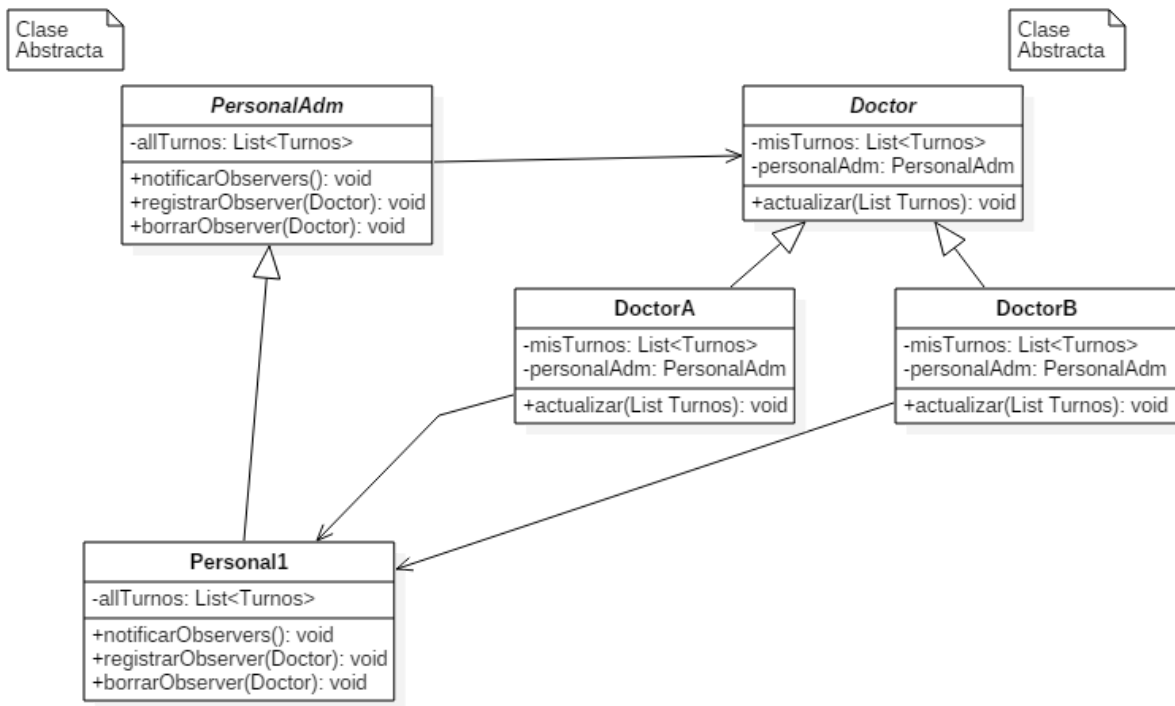
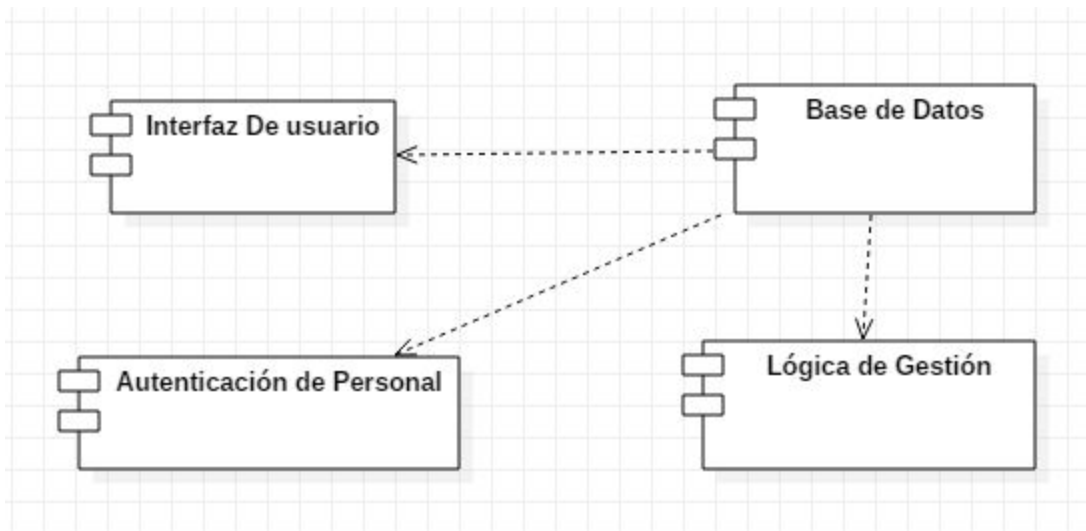
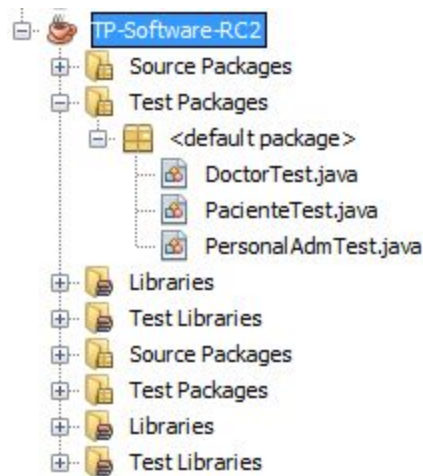


Diagrama de Despliegue



Pruebas unitarias automáticas

Las pruebas fueron realizadas con JUnit a las clases Doctor, Paciente y PersonalAdm, para poder cubrir las principales actividades de nuestro sistema. Dichas pruebas se encuentran dentro del paquete Test como se muestra a continuación.



Para poder correrlas, se debe ejecutar Run File a la clase que queramos testear e inmediatamente se nos mostrará qué métodos de la clase pasaron la prueba y cuáles no.

Tests passed: 100,00 %

All 6 tests passed. (0,28 s)

- ✓ PacienteTest passed
 - ✓ testGetAlergias passed (0,003 s)
 - ✓ testPagar passed (0,0 s)
 - ✓ testGetMisMedicamentos passed (0,0 s)
 - ✓ testAddAlergia passed (0,001 s)
 - ✓ testAddMedicamento passed (0,001 s)
 - ✓ testSetMifdp passed (0,0 s)

Tests passed: 100,00 %

All 4 tests passed. (0,45 s)

- ✓ PersonalAdmTest passed
 - ✓ testCrearPaciente passed (0,003 s)
 - ✓ testBorrarTurno passed (0,002 s)
 - ✓ testCrearTurno passed (0,0 s)
 - ✓ testCobrar passed (0,0 s)

Tests passed: 100,00 %

All 4 tests passed. (0,279 s)

- ✓ DoctorTest passed
 - ✓ testRecetar passed (0,003 s)
 - ✓ testGetMisPacientes passed (0,002 s)
 - ✓ testAgregarAlergia passed (0,0 s)
 - ✓ testGetTurnos passed (0,0 s)

Como podemos ver, las principales funciones de nuestro sistema están testeadas y aprobadas.